2.8 ONBOARD DECEPTIVE ECM

Several discrepancies were found in the Onboard Deceptive ECM Functional Element (FE) for ESAMS 2.7. The overall code quality is good, however, numerous corrections are recommended for the internal documentation. External documentation for ESAMS 2.7 is generally good, but is sometimes incomplete and/or incorrect in its presentation of ECM theory and methodology.

Table 2.8-1 listed below summarizes the desk-checking and software testing verification activities for each design element in the Onboard Deceptive ECM FE. A design element is an algorithm that represents a specific component of the FE design. One entry is listed for each design element. The two results columns contain checks if no discrepancies were found during verification. Where discrepancies were found, the desk check results column contains references to discrepancies listed in Table 2.8-4, while the test case results column lists the number of the relevant test case in Table 2.8-6. More detailed information on the results is recorded in these tables.

TABLE 2.8-1. Verification Results Summary.

DESIGN ELEMENT	CODE LOCATION	DESK CHECK RESULT	TEST CASE ID	TEST CASE RESULT
8-1 Terrain Bounce Jammer Leakage	BEMGRM 249-405 BEMSVL BEMEXC BEMOUT	4	8-1	4
8-2 Calculation of Amplitude Modulation Jammer on and off Times	BEMEXC GETWOB	D1 D2	8-2 to 8-4	8-2 to 8-4
8-3 Complex Voltage from Amplitude Modulation Jamming	BEMGRM 249-405 BEMTVL BEMEXC 79-109 BEMOUT	4	8-5	4
8-4 Terrain Bounce Geometry	ATJCON 222-260 ATJREF	D3	8-6	8-6
8-5 Power Calculations With Perfect Reflection	ATJCON 261-266 ATJBOR ATJMPI	4	8-7	4

TABLE 2.8-1. Verification Results Summary. (Contd.)

DESIGN ELEMENT	CODE LOCATION	DESK CHECK RESULT	TEST CASE ID	TEST CASE RESULT
8-6 Terrain Bounce Path Loss Calculations	ATJFRC ATJRSC ATJSDC ATJCON 267-278	D4	8-8	8-8
8-7 Phase, Doppler and Range Calculations	ATJSIG ATJDGP ATJCON 287-288	D5	8-9	8-9
8-8 Complex Voltage from Terrain Bounce Jamming	ATJSIG ATJDGP ATJCON 279-294	4	8-10	4
8-9 Complex Voltage from Cross-eye Jamming	BEMGRM BEMSEN BEMTVL BEMEXC BEMOUT BEMSET	4	8-11	4
8-10 Complex Voltage from VGPO, RGPO, and RVGPO Jamming	BEMGRM BEMSEN BEMTVL BEMEXC BEMOUT BEMSET	4	8-12 to 8-14	4

2.8.1 Overview

The objective of Deceptive ECM is to mask the real target signal by injecting suitably modified replicas of the real target signal (i.e., generating false targets) into the victim system. Deceptive jamming techniques can degrade angle, range, or doppler tracking. With the latter two, there is no angle deception, so the target could still be intercepted if some evasive maneuver is not taken after the jam signal is dropped. Deceptive jamming is typically used after a threat has acquired and locked on to the target and is in the process of firing ordinance at it. Onboard Deceptive ECM involves the generation of deceptive jamming signals from sources originating on the intended target itself as opposed to those originating from off-board locations. There are four different types of deception jamming: amplitude modulation (AM), terrain bounce (TB), cross-eye (CE), and gate pulloff. The Onboard Deceptive ECM Functional Element for ESAMS models all these types of deception jamming with the last type implemented using three different techniques, range gate pulloff (RPGO), velocity gate pulloff (VGPO), or the coordinated combination of the two (RVGPO).

ESAMS 2.7 implementation of Onboard Deceptive ECM is accomplished with nine general and eleven designated subroutines. Subroutines BEMGRM, BEMSEN, BEMANT, BEMTVL, BEMSVL, BEMEXC, BEMOUT and BEMSET can be used by all techniques in the Onboard Deceptive ECM FE. None of these subroutines are exclusively designated for this FE and are shared with other FE's that implement ECM techniques. However, these subroutines are exclusively designated for ECM techniques in general, with Onboard Deceptive being one of six ECM FE categories modeled by ESAMS. The subroutine GETWOB is used only for a special case of the amplitude modulation technique called "wobbulation". The subroutines ATJREF, ATJBOR, ATJMPI, ATJFRC, ATJRSC, ATJSDC, ATJSIG, ATJDGP, and ATJCON are used exclusively by the terrain bounce technique. No special coding was used to implement either the cross-eye or gate pulloff techniques, with these methods being implemented by waveform setup in the ECMD file. In addition, ECMINI and ATJI are higher level subroutines that are used to initialize ECM techniques, with ATJI being used exclusively by the terrain bounce technique. The twenty subroutines used for this FE are described in Table 2.8-2.

TABLE 2.8-2. Onboard Deceptive ECM Subroutine Descriptions.

MODULE NAME	DESCRIPTION
BEMGRM	Checks each technique in the ECMD file to see if it is active at the current time against the current radar. Serves as top level routine for ECM calculations.
BEMSEN	Sets up engagement features between the jamming aircraft and the ground radar, missile seeker, or missile fuze.
BEMANT	Provides jamming antenna position, velocity, and orientation.
BEMSVL	Calculates relative geometries and orientations between the missile seeker and jamming aircraft.
BEMTVL	Calculates relative geometries and orientations between the ground radar and jamming aircraft.
BEMEXC	Loads the current ECM characteristics. These include doppler, power, phase, polarization, pulse width, and time delay.
BEMOUT	Develops the ECM-induced voltage in the victim radar receiver.
BEMSET	Sets flags for printing event message output. Each ECM signal is examined for turning on or off.
ECMINI	Initializes ECM simulations. Provides initialization for both noise and ECM techniques.
GETWOB	Determines if a wobbulation sweep pulse is on or off at the current time.
ATJI	Initializes some of the angle-track jamming functions.
ATJREF	Calculates geometric relationships.
ATJBOR	Get decoy angles off-boresight.
ATJMPI	Calculates decoy power at missile without terrain effects.
ATJFRC	Calculates Fresnel reflection coefficient.
ATJRSC	Calculates Rayleigh scattering coefficient.
ATJSDC	Calculates spatial distribution coefficient.
ATJSIG	Calculates decoy sum and difference voltages.
ATJDGP	Calculates decoy doppler frequency.
ATJCON	Passes terrain bounce data to BEMGRM.

2.8.2 Verification Design Elements

The ten design elements defined for the Onboard Deceptive ECM FE are listed in Table 2.8-3; they are fully described in Section 2.8.2 of ASP-II. A design element is an algorithm that represents a specific component of the FE design. Design elements 8-1 through 8-10 model the four types of Onboard Deceptive ECM used by ESAMS 2.7.

TABLE 2.8-3. Onboard Deceptive ECM Design Elements.

SUBROUTINE	DESIGN ELEMENT	DESCRIPTION
BEMGRM BEMSEN BEMANT BEMSVL BEMEXC BEMOUT	8-1 Terrain Bounce Jammer Leakage	This ECM technique attempts to generate a false target by bouncing a repeater jamming signal off of the terrain which reflects it to the missile seeker receiver. This design element models the jammer leakage signal in the direction of the missile seeker. Since this signal propagates directly to the missile instead of bouncing off the ground as intended, it can inadvertently aid the missile in intercepting the target.
GETWOB	8-2 Calculation of Amplitude Modulation Jammer on and off Times	This ECM technique attempts to induce angle track errors into centroidal tracking (TWS or conical scan) radars. It can also impact the target tracking capability of monopulse radars through exploitation of channel imbalance. This design element addresses the calculation of the amplitude modulation frequency of the jammer.
BEMGRM BEMSEN BEMANT BEMTVL BEMEXC BEMOUT	8-3 Complex Voltage from Amplitude Modulation Jamming	This ECM technique attempts to induce angle track errors into centroidal tracking (TWS or conical scan) radars. It can also impact the target tracking capability of monopulse radars through exploitation of channel imbalance. This design element models the complex amplitude modulation jamming voltages for the sum, azimuth difference and elevation difference channels in the victim radar receiver.
ATJCON ATJREF	8-4 Terrain Bounce Geometry	This ECM technique attempts to generate a false target by bouncing a repeater jamming signal off of the terrain which reflects it to the missile seeker receiver. This design element models the geometry of the apparent target signal (the decoy) as seen by the missile seeker.
ATJBOR ATJMPI	8-5 Power Calculations With Perfect Reflection	This design element is part of the terrain bounce technique in that it initially models perfect reflection of the jamming signal without the effects of terrain. The effects of the terrain are accounted for at a later time.
ATJCON ATJFRC ATJRSC ATJSDC	8-6 Terrain Bounce Path Loss Calculations	This design element models the Fresnel reflection coefficient and the Rayleigh scattering coefficient in order to determine the energy lost during transmission into the terrain and scattering due to terrain roughness. It also models the spatial distribution coefficient in order to determine the impact that diffuze reflection has on seeker performance.
ATJSIG ATJDGP ATJCON	8-7 Phase, Doppler and Range Calculations	The calculation of these parameters is more complex for the terrain bounce technique due to the decoy signal being reflected from the ground. These calculations have to take into account the path from the target to the decoy and that from the decoy to the missile, instead of just the direct missile-target path as with other techniques.

TABLE 2.8-3. Onboard Deceptive ECM Design Elements. (Contd.)

SUBROUTINE	DESIGN ELEMENT	DESCRIPTION
ATJSIG ATJDGP ATJCON	8-8 Complex Voltage from Terrain Bounce Jamming	This design element models the complex terrain bounce jamming voltages for the sum, azimuth difference and elevation difference channels in the victim radar receiver.
BEMGRM BEMSEN BEMANT BEMTVL BEMEXC BEMOUT	8-9 Complex Voltage from Cross-eye Jamming	This ECM technique uses two jamming signals of equal amplitude, but 180 degrees out-of-phase in an attempt to induce tracking errors into the victim radar. This is accomplished by distorting the wavefront reaching the victim radar which attempts to align its antenna normal to the incoming signal wavefront. When this wavefront is distorted, tracking angles can develop. Returns sum and difference channel voltages in the victim radar receiver due to these jamming signals.
BEMGRM BEMSEN BEMANT BEMTVL BEMEXC BEMOUT	8-10 Complex Voltage from VGPO, RGPO, and RVGPO Jamming	This technique involves gate stealing by pulling the velocity gate, the range gate, or both off of the target and then dropping the jam signal. If successful, the radar goes into a coast mode and attempts reacquisition. A prompt maneuver by the target may be effective in degrading missile intercept capability. Returns sum and difference channel voltages in the victim radar receiver due to gate stealing ECM.

2.8.3 Desk Checking Activities and Results

The code implementing this FE was manually examined using the procedures described in Section 1.1 of this report. Any discrepancies discovered are described below in Table 2.8-4.

TABLE 2.8-4. Code Discrepancies.

DESIGN ELEMENT	DESK CHECK RESULT
8-2 Calculation of Amplitude Modulation Jammer On and Off Times	D1. The up and down slopes for the sawtooth sweep type, SLOPE on lines 71 and 75 of GETWOB, are only half of what they should be according to ASP II Equations [2.8-10] and [2.8-12]. D2. The equations for the calculation of the on-time for the amplitude modulation pulse have been incorrectly implemented. Equations used to calculate the variables DISCR, the discriminant for calculation of AM frequency time steps, and ZROTIM, the AM pulse on-time, need to be revised to accurately reflect ASP II Equations [2.8-14] through [2.8-16]. The current implementation results in an incorrect number of counts to complete a pulse. This also results in problems when a sawtooth sweep switches from a positive slope to a negative slope at the sweep period midpoint. The stop frequency of the upsweep reaches only f_c, instead of $f_c + \Delta f$, before switching to a downsweep. If the slopes for the sawtooth sweep are corrected as recommended in deficiency D1, the stop frequency of the upsweep is correct, $f_c + \Delta f$, but the start frequency of the downsweep is shifted downward by $2 \Delta f$. This results in the downsweep starting at a frequency of $f_c - \Delta f$ instead of at $f_c + \Delta f$ as it should.
8-4 Terrain Bounce Geometry	D3. When the decoy elevation angle, CMDPE is calculated on lines 115 and 148 of the subroutine ATJREF, a one is added to the target-decoy slant range, CMTDR in the denominator of the argument. This was probably done to prevent a singularity condition that would result when trying to calculate arcsine with an argument greater than one. Rather than adding in a fudge factor to prevent this problem, it would be preferable to calculate the argument without adding one to the denominator. The value of the argument should then be compared to one using an AMIN1 statement, with CMDPE using the minimum value of the two for its argument. This would limit the argument for CMDPE to a maximum of one in case it was actually calculated to be greater than one. Otherwise, this angle should be calculated having only the target-decoy slant range in the denominator of its argument in order to maximize it’s accuracy.

TABLE 2.8-4. Code Discrepancies. (Contd.)

DESIGN ELEMENT	DESK CHECK RESULT
8-5 Terrain Bounce Power Calculations with Perfect Reflection	D4. This discrepancy is essentially the same problem described in discrepancy D3. When the decoy elevation and azimuth angles in the jammer antenna frame, CMDAE and CMDAA, are calculated on lines 64 and 65 of the subroutine ATJBOR, a one has been added to the denominator in each argument. In the case of CMDAE, it was added to the target-decoy slant range, CMTDR, and in the case of CMDAA, it was added to the decoy x-dimension, XAD. This was probably done to prevent a singularity condition that would result when trying to calculate either arcsine or arctangent with an argument greater than one. Rather than adding in a fudge factor to prevent this problem, it would be preferable to calculate the argument without adding one to the denominator. The value of the argument should then be compared to one using an AMIN1 statement, with CMDAE or CMDAA, whatever the case may be, using the minimum value of the two for its argument. This would limit the arguments for CMDAE and CMDAA to a maximum of one in case they were actually calculated to be greater than one. Otherwise, these angles should be calculated without having the extra one added to the denominator of their arguments in order to maximize the accuracy of these equations.
8-6 Terrain Bounce Path Loss Calculations	D5. This discrepancy is essentially the same problem described in discrepancies D3 and D4. When the target-missile altitude, HTHM, is calculated on line 69 of the subroutine ATJSDC, a one has been added to the missile position z-component, Z in the denominator of the equation. This was probably done to prevent the target-missile altitude ratio from exceeding one. This, in turn, could prevent the spatial distribution coefficient from exceeding one. Rather than adding in a fudge factor to prevent this problem, it would be preferable to calculate this ratio without adding one to the denominator. The value of this ratio should then be compared to one using an AMIN1 statement, with HTHM using the minimum value of the two. This would limit the value of HTHM to a maximum of one in case it was actually calculated to be greater than one. Otherwise, this ratio should be calculated without having the extra one added to the denominator in order to maximize the accuracy of the equation.

Except as noted in Table 2.8-5 below, overall code quality was evaluated as good and internal documentation was evaluated as fair. In most cases, subroutine I/O and logical flow were found to match the CMS descriptions.

.

TABLE 2.8-5. Code Quality and Internal Documentation Results.

SUBROUTINE	CODE QUALITY	INTERNAL DOCUMENTATION
ECMINI	INCLUDE statements for the common blocks FLAGS, GRADAR, and PROGC are not necessary because the variables contained in them are not used.	The subroutine contains no version number. The source code shows this subroutine to be 'UNCLASSIFIED', when in fact it is actually classified as 'SECRET'. The variables IERRIP, IPT, JPT, ISPS, ISPS5, MODE, NUMWDS, and PATFIL are missing from the list of local variables. The indices I and J are in this list but are not used. The variables IFATAL and IWARN are missing from the parameters list. The variables IACQR, ILUMR, ISEKR, ITRKR, MRADFL, and NUMJAM are in this list but is not used. The variables ANSLW, CHRPT, CNTFRQ, DUTCYL, ECMT, LPATRN, OFFFRQ, PANT, RMPTIM and SWPTYP are missing from the list of variables for the common block ECMD. JCHFRC is in this list of variables for the common block ECMI but is not used. GRADAR and PROGC are in the list of common blocks but their respective variables WVLTX and CHFRC are not used. The common blocks ECMV, PROGVI and RUNVI with their respective variables TIMMOD, ISPS(5) and LUNLP are missing from the list of common blocks. The subroutines CKTLU2, ERROR, and RDF along with the library function NINT are missing from the list of subroutines called. The subroutine CHAFFI is in this list but is not used. The comments on lines 140, 145, and 160 that refer to the transmit antenna are incorrect. These comments should be referring to the receive antenna instead.
BEMGRM	The INCLUDE statements on lines 212 and 220 for the common blocks ECCHAF and PROGC are not necessary since the variables they contain are not used.	The definition of the local variable PSIBM is wrong. It should be defined as the target yaw angle, not the target roll angle. The variable RADCHF is missing from the list of local variables.

TABLE 2.8-5. Code Quality and Internal Documentation Results. (Contd.)

SUBROUTINE	CODE QUALITY	INTERNAL DOCUMENTATION
BEMSEN	OK	The definitions of the calling arguments XVJ, YVJ, and ZVJ are wrong. They should be defined as the X, Y, and Z components of the victim site-to-jammer antenna vector. The variables ANTPHI, ANTTHT, and ANTPSI are missing from the list of local variables. The variables AMISX, AMISY, AMISZ, FUZX, FUZY, FUZZ, XSJ, YSJ, and ZSJ are in this list but are not used. The subroutines AFMPOS, MISPOS, and SITPOS are in the list of subroutines called by BEMSEN but are not used.
BEMTVL	The INCLUDE statement on line 71 for the common block FREND is not necessary since the variables that it contains are not used.	The variables ANTPHI, ANTTHT, and ANTPSI are missing from the list of calling arguments. The variables ANULL and RGAIN are missing from the list of local variables. The variable ILUMR is missing from the list of parameters. The subroutine ANTGAN is missing from the list of subroutines called by BEMTVL. The comment on line 103 for the variable RADVLU(2) is incorrect. This equation is actually for the power density at the target.
BEMSVL	The INCLUDE statements on lines 126 and 127 for the common blocks FLAGS and FREND are not necessary since the variables they contain are not used.	The variables ANTPHI, ANTTHT, and ANTPSI are missing from the list of calling arguments. The variables ANULL and RGAIN are missing from the list of local variables. The variables RANGE, XAT, YAT, ZAT, XSAT, YSAT, and ZSAT are in this list but are not used. FLAGS and FREND are in the list of common blocks but the variables they contain are not used. The variable ALPOFF is in the list of variables for the common block GRADAR but is not used. The subroutine ANTGAN is missing from the list of subroutines called by BEMSVL. The comment on line 183 for the variable RADVLU(2) is incorrect. This equation is actually for the power density at the target.

TABLE 2.8-5. Code Quality and Internal Documentation Results. (Contd.)

SUBROUTINE	CODE QUALITY	INTERNAL DOCUMENTATION
BEMANT	INCLUDE statements for ARYBND and PARAM are not necessary since the parameters contained in these common blocks are not used. Missing variable declaration statements for IANT, IONDCY, ITCHNQ, TIMEB, ANTX, ANTY, ANTZ, ANTXD, ANTYD, ANTZD, ANTPSI, ANTTHT, ANTPHI, TGTPSI, TGTTHT, TGTPHI, TGTX, TGTY, TGTZ, TGTXD, TGTYD and TGTZD need to be added to the code.	The variables ANTPHI, ANTTHT, and ANTPSI are missing from the list of calling arguments. The library function NINT and the subroutine TDROLL are missing from the list of subroutines called by BEMANT.
BEMEXC	OK	The variable IONFLG is missing from the list of local variables. The library functions NINT and FLOAT along with the subroutine GETWOB are missing from the list of subroutines called by BEMEXC.
BEMOUT	The calls to the subroutine GYRATE on lines 137 and 147 are redundant and can be consolidated into one call. This can be accomplished by moving the call to line 131 so that it occurs before the IF statement for the fixed or slewable antenna. The functions ASIN and ATAN2 on lines 141 and 143, respectively, can be replaced by the functions ASIND and ATAND2 which return answers in degrees, not radians. This would eliminate the need to convert them to degrees by multiplying them by the radians-to-degrees conversion factor R2D.	Comments for the calculation of the variables ECHVLT and ACHVLT state that these are difference channel gains, when in fact they are difference channel voltages. The variables IDBUS, IXPNT1, IXPNT2, KODAMP, NUMPRO, PCBWJM, SPCWID, and XMTPAT are missing from the list of calling arguments. The variables I2B, KFREQ, KPWR, KPHASE, KPWID, KRVGAZ, KRVGEL, KRVSUM, KRVVCL, and KTDEL are missing from the list of parameters. The variable PIX2 is in this list but is not used. The subroutines BEMNZ and TGTROL are missing from the list of subroutines called by BEMOUT. The library function AMOD is in this list but is not used.
BEMSET	OK	The variable NUMTEC is in the calling argument list for BEMSET but is not used.
GETWOB	The INCLUDE statements on lines 41 and 42 for the common blocks PARAM and ARYBND are not necessary since the variables they contain are not used.	The version number of this subroutine is missing from the comments. The library functions ABS, FLOAT, NINT, and SQRT are missing from the list of subroutines called by GETWOB.
ATJREF	The INCLUDE statements on lines 93 and 94 for the common blocks PARAM and ARYBND are not necessary since the variables they contain are not used.	The version number and the date last modified for this subroutine are missing from the comments. The subroutine GTTTD is missing from the list of subroutines called by ATJREF.
ATJBOR	OK	The version number and the date last modified for this subroutine are missing from the comments.

TABLE 2.8-5. Code Quality and Internal Documentation Results. (Contd.)

SUBROUTINE	CODE QUALITY	INTERNAL DOCUMENTATION
ATJMPI	OK	The version number and the date last modified for this subroutine are missing from the comments.
ATJFRC	OK	The version number and the date last modified for this subroutine are missing from the comments.
ATJRSC	OK	The version number and the date last modified for this subroutine are missing from the comments.
ATJSDC	OK	The version number and the date last modified for this subroutine are missing from the comments.
ATJSIG	The INCLUDE statements on lines 75 and 78 for the common blocks ARYBND and GRADAR are not necessary since the variables they contain are not used.	The version number and the date last modified for this subroutine are missing from the comments. The variables ICALC, ICIRC, PGIMAN, ROTB, and YGIMAN are not used in this subroutine and should be deleted from the argument description list. The variables GSUM, GDIFAZ, and GDIFEL are not among the calling arguments for this subroutine, therefore, their descriptions should be moved to the local variables list. The variables AZOB and ELOB are not used in this subroutine and should be deleted from the local variables list.
ATJDGP	OK	The version number and the date last modified for this subroutine are missing from the comments. The word ‘none’ should be inserted under the heading for subroutines called.

TABLE 2.8-5. Code Quality and Internal Documentation Results. (Contd.)

SUBROUTINE	CODE QUALITY	INTERNAL DOCUMENTATION
ATJCON	The INCLUDE statements on lines 181 and 187 for the common blocks PARAM, FREND, MSLD, MSLTGT, and RDRD are not necessary since the variables they contain are not used.	The variable JAMINX is not used in this subroutine and should be deleted from the argument description list. The variables SGSV, SGDVA, SGDVE, RTSI, SGDOP, SGPW, and NUMPRO are arguments for this subroutine and are not passed on through common blocks. Descriptions of these variables should be moved to the argument list and any references to the common blocks SIGNLC, SIGNLI, and SIGNLR should be removed. In addition, the variables SGPCBW and IDBUS should also be described in the list of calling arguments. All of the variables that are currently in the list of calling arguments, APOYNT, DOPPLR, EPOYNT, IANTEN, POWER, PWIDTH, and TDEL, have been defined as unknown. Definitions for these variables should be incorporated in this list. No list of local variables currently exists in the comments for this subroutine. One should be developed to include variable descriptions for XR, YR, ZR, XRDOT, YRDOT, ZRDOT, RTR2, and PATH. The variable E is not used in this subroutine and should be deleted from the parameter list. The variable ANGOB is not used in this subroutine and should be deleted from the variable list for the common block ECMV. The variable NOUT is not used in this subroutine and should be deleted from the variable list for the common block FLAGS. The library function DB is not called by this subroutine and should be deleted from the list of subroutines called.
ATJI	The INCLUDE statements on lines 66, 67, 68, 70, and 73 for the common blocks CONST, ARYBND, PARAM, ECMD and SIMVR are not necessary since the variables they contain are not used by this subroutine.	The variables CMJLFD, CMPHDT, and TNOA are not used in this subroutine and should be deleted from the list of local variables. The variable SPDLGT is not used in this subroutine (nor are any other parameters) and should be deleted along with the parameter list. Variable lists for the common blocks ECMD and SIMVR can be deleted since none from them are used. The variable NOA is not used in this subroutine and should be deleted from the variable list for the common block ECMI. The variable HPANG should be added to the list for the common block FREND. The library function MAX should be added to the list of subroutines called.

2.8.4 Software Test Cases

All subroutines implementing the Onboard Deceptive ECM functional element were tested using integrated code. Since many of the subroutines that comprise this FE have already been tested during verification activities for the Onboard Noise ECM FE, software testing for this FE was focused on the ten subroutines designated for the terrain bounce technique (ATJ*) and the subroutine GETWOB for the amplitude modulation technique. For the ECM techniques that had no unique lines of code, test cases were run using the appropriate ECM data files to verify that no overflows or singularity conditions developed that would cause fatal program errors. However, detailed calculations for each step of these techniques were not performed since these subroutines had been previously tested during verification of the Onboard Noise ECM FE. For integrated testing, the entire ESAMS model was run in debug mode. The standard ESAMS data files for the systems under consideration were used as input for all test cases.

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE.

Test Case ID	TEST CASE DESCRIPTION
8-1	OBJECTIVE: Verify correct calculation of the direct path complex voltages due to jammer leakage in the direction of the missile seeker for the terrain bounce ECM technique. PROCEDURE: 1. Run ESAMS, and observe in Subroutine BEMSVL the values of XJST, YJST, and ZJST. 2. Continue execution, and observe the values of XJSTD, YJSTD, and ZJSTD. 3. Continue execution, and observe the values of RJST and VCLJ. 4. Continue execution, and observe the values of XJM, YJM, and ZJM. 5. Continue execution, and observe the values of XJMD, YJMD, and ZJMD. 6. Continue execution, and observe the value of RADVLU(1). 7. Continue execution, and observe the values of RTS, RADVLU(2), and RADVLU(3). 8. Continue execution, and observe the values of RADVLU(4), RADVLU(5), RJM, RADVLU(9), RADVLU(10), RADVLU(11), RADVLU(12), and RADVLU(13). 9. Continue execution, and observe in Subroutine BEMEXC the six values of the array VALUE. 10. Continue execution, and observe in Subroutine BEMGRM the values of WAVLEN, ERAZ, EREL, and XLS. 11. Continue execution, and observe in Subroutine BEMOUT the values of NUMPRO, SUMJAM, DFJMAZ (after line 190), DFJMEL (after line 192), RSJAM, DOPJAM, PWJAM, and PCBWJM. VERIFY: 1. The values of XJST, YJST, and ZJST in step 1 equal 6389.48, 0, and 148, respectively. 2. The values of XJSTD, YJSTD, and ZJSTD in step 2 equal -250, 0, and 0, respectively. 3. The values of RJST and VCLJ in step 3 equal 6391.1938 and -249.9330, respectively. 4. The values of XJM, YJM, and ZJM observed in step 4 equal 333.133, 0, and -32.3801. 5. The values of XJMD, YJMD, and ZJMD observed in step 5 equal -1762.095, 0, and 189.854. 6. The values of RADVLU(1) observed in step 6 equals 8336.8662. 7. The values of RTS, RADVLU(2), and RADVLU(3) observed in step 7 equal 6391.1938, 3.0172, and 1.7528. 8. The values of RADVLU(4), RADVLU(5), RJM, RADVLU(9), RADVLU(10), RADVLU(11), RADVLU(12), and RADVLU(13) observed in step 8 equal 2.1575×10^{-4}, 7.47×10^{-7}, 334.703, -1772.1965, 40.07626, 3.0634436×10^{-3}, 6.05037, and 4.233502. 9. The values of VALUE(1), VALUE(2), VALUE(3), VALUE(4), VALUE(5), and VALUE(6) observed in step 9 equal 8336.0136, 100, 1.7528, 2.1575×10^{-4}, 7.47×10^{-7}, and 0. 10. The values of WAVLEN, ERAZ, EREL, and XLS observed in step 10 equal 0.0300, $-6.3727103 \times 10^{-2}$, 6.3727103×10^{-2}, and 2.9854, respectively. 11. The values of NUMPRO, SUMJAM, DFJMAZ, DFJMEL, RSJAM, DOPJAM, PWJAM, and PCBWJM observed in step 11 equal 2, (-7.1968×10^{-6}, 2.5989×10^{-5}), (2.3815×10^{-8}, -8.6×10^{-8}), (-1.1109×10^{-6}, 4.0116×10^{-6}), 334.7030, 67450.1258, 7.47×10^{-7}, and 0. RESULT: OK

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	TEST CASE DESCRIPTION
8-2	OBJECTIVE: Check amplitude modulation on and off time calculations in Subroutine GETWOB for the upsweep case. PROCEDURE: 1. Start ESAMS and observe in Subroutine BEMEXC the values of SWPTYP, ITCHNQ, TMPTIM, PERIOD, and JTCMOD. 2. Observe the value of SLOPE in the Subroutine GETWOB. 3. Observe the value of STRFRQ in the Subroutine GETWOB. 4. Observe the values of ICOUNT and DISCR in the Subroutine GETWOB. 5. Observe the value of TIMON in the Subroutine GETWOB. 6. Observe the value of TOFF in the Subroutine GETWOB. 7. Observe the values of IONFLG and TIMBE4 in the Subroutine GETWOB. 8. Continue execution until call to BEMEXC at first time step after TOFF and observe the value of TMPTIM. 9. Continue execution and observe the value of IONFLG at the end of the Subroutine GETWOB right before the return to BEMEXC. 10. Calculate start frequency of frequency sweep by taking the inverse of two times TOFF. 11. Continue execution until last call to BEMEXC before end of frequency sweep at time = 4 sec and observe values of ICOUNT, TIMON, and TOFF in the Subroutine GETWOB. 12. Calculate stop frequency of frequency sweep by taking the inverse of two times TOFF. VERIFY: 1. The values observed in step 1 equal 1, 1, 0, 6, and 0. 2. The value observed in step 2 equals independent calculation of ASP II Equation [2.8-6]. 3. The value observed in step 3 equals independent calculation of ASP II Equation [2.8-7]. 4. The values observed in step 4 equal 0 and the squared result of independent calculation of ASP II Equation [2.8-14]. 5. The value observed in step 5 equals independent calculation of ASP II Equation [2.8-15]. 6. The value observed in step 6 equals independent calculation of ASP II Equation [2.8-17]. 7. The values observed in step 7 equal 1 and 0. 8. The value observed in step 8 is slightly larger than TOFF (0.0069). 9. The value observed in step 9 equals 0. 10. The value calculated in step 10 equals STRFRQ (72). 11. The values observed in step 11 equal ICOUNT = 336 and independent calculation of ASP II Equations [2.8-15] and [2.8-17]. 12. The value calculated in step 12 equals CTRFRQ + OFFFRQ (84). RESULT: The calculated value of DISCR in step 4 is half of what independent calculations produce. The last count before the end of the sweep period at t = 4 sec was observed as 311 in step 11, while a total pulse count of 336 was calculated by hand. Due to this count discrepancy, the pulse on and off time calculations for TIMON and TOFF don't agree with those calculated using the ASP II equations at the end of the sweep period at time = 4 seconds. The formulation of the equations for the variable ZROTIM account for the first discrepancy and result in start and stop times that agree with ASP II Equations [2.8-15] and [2.8-17], at least for calculations at low counts. However, there was no compensation for the second discrepancy regarding the number of counts required to reach the pulse off-time being different from those calculated using the ASP II equations.

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	TEST CASE DESCRIPTION
8-3	OBJECTIVE: Check amplitude modulation on and off time calculations in Subroutine GETWOB for the downsweep case. PROCEDURE: 1. Start ESAMS and observe in Subroutine BEMEXC the values of SWPTYP, ITCHNQ, TMPTIM, PERIOD, and JTCMOD. 2. Observe the value of SLOPE in the Subroutine GETWOB. 3. Observe the value of STRFRQ in the Subroutine GETWOB. 4. Observe the values of ICOUNT and DISCR in the Subroutine GETWOB. 5. Observe the value of TIMON in the Subroutine GETWOB. 6. Observe the value of TOFF in the Subroutine GETWOB. 7. Observe the values of IONFLG and TIMBE4 in the Subroutine GETWOB. 8. Continue execution until call to BEMEXC at first time step after TOFF and observe the value of TMPTIM. 9. Continue execution and observe the value of IONFLG at the end of the Subroutine GETWOB right before the return to BEMEXC. 10. Calculate start frequency of frequency sweep by taking the inverse of two times TOFF. 11. Continue execution until last call to BEMEXC before end of frequency sweep at time = 4 sec and observe values of ICOUNT, TIMON, and TOFF in the Subroutine GETWOB. 12. Calculate stop frequency of frequency sweep by taking the inverse of two times TOFF. VERIFY: 1. The values observed in step 1 equal 2, 1, 0, 6, and 0. 2. The value observed in step 2 equals independent calculation of ASP II Equation [2.8-8]. 3. The value observed in step 3 equals independent calculation of ASP II Equation [2.8-9]. 4. The values observed in step 4 equal 0 and the squared result of independent calculation of ASP II Equation [2.8-14]. 5. The value observed in step 5 equals independent calculation of ASP II Equation [2.8-16]. 6. The value observed in step 6 equals independent calculation of ASP II Equation [2.8-17]. 7. The values observed in step 7 equal 1 and 0. 8. The value observed in step 8 is slightly larger than TOFF (0.0060). 9. The value observed in step 9 equals 0. 10. The value calculated in step 10 equals STRFRQ (84). 11. The values observed in step 11 equal ICOUNT = 288 and independent calculation of ASP II Equations [2.8-16] and [2.8-17]. 12. The value calculated in step 12 equals CTRFRQ - OFFFRQ (72). RESULT: The calculated value of DISCR in step 4 is half of what independent calculations produce. The last count before the end of the sweep period at t = 4 sec was observed as 311 in step 11, while a total pulse count of 288 was calculated by hand. Due to this count discrepancy, the pulse on and off time calculations for TIMON and TOFF don’t agree with those calculated using the ASP II equations at the end of the sweep period at time = 4 seconds. The formulation of the equations for the variable ZROTIM account for the first discrepancy and result in pulse start and stop times that agree with ASP II Equations [2.8-16] and [2.8-17], at least for calculations at low counts. However, there was no compensation for the second discrepancy regarding the number of counts required to reach the pulse off-time being different from those calculated using the ASP II equations.

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	TEST CASE DESCRIPTION
8-4	OBJECTIVE: Check amplitude modulation on and off time calculations in Subroutine GETWOB for the case of a sawtooth sweep type. PROCEDURE: 1. Start ESAMS and observe in Subroutine BEMEXC the values of SWPTYP and TMPTIM. 2. Observe the values of SLOPE and STRFRQ in the Subroutine GETWOB. 3. Observe the values of ICOUNT and DISCR in the Subroutine GETWOB. 4. Observe the values of TIMON and TOFF in the Subroutine GETWOB. 5. Observe the values of IONFLG and TIMBE4 in the Subroutine GETWOB. 6. Continue execution until call to BEMEXC at first time step is greater than TOFF and observe the value of TMPTIM. 7. Continue execution and observe the value of IONFLG at the end of the Subroutine GETWOB. 8. Calculate start frequency of the upsweep by taking the inverse of two times TOFF. 9. Continue execution until last call to BEMEXC before sweep period midpoint at time = 2 sec and observe values of ICOUNT, TIMON, and TOFF in the Subroutine GETWOB. 10. Calculate stop frequency of sawtooth upsweep by taking the inverse of two times TOFF. 11. Continue execution until first call to BEMEXC after sweep period midpoint at t = 2 sec and observe values of SLOPE, STRFRQ, ICOUNT, TIMON, and TOFF in Subroutine GETWOB. 12. Calculate start frequency of sawtooth downsweep by taking the inverse of two times TOFF. 13. Continue execution until last call to BEMEXC before end of frequency sweep at time = 4 sec and observe values of ICOUNT, TIMON, and TOFF in the Subroutine GETWOB. 14. Calculate stop frequency of sawtooth downsweep by taking the inverse of two times TOFF. VERIFY: 1. The values observed in step 1 equal 3 and 0. 2. The values observed in step 2 equals independent calculation of ASP II Equations [2.8-10 & 11]. 3. The values observed in step 3 equals 0 and matches the squared result of independent calculation of ASP II Equation [2.8-14], respectively. 4. The value observed in step 4 equals independent calculation of ASP II Equations [2.8-15 & 17]. 5. The values observed in step 5 equal 1 and 0. 6. The value observed in step 6 is slightly larger than TOFF (0.0069). 7. The value observed in step 7 equals 0. 8. The value calculated in step 8 equals STRFRQ (72). 9. The values observed in step 9 equal 167 and independent calculation of ASP II Equations [2.8-15] and [2.8-17], respectively. 10. The value calculated in step 10 equals CTRFRQ + OFFFRQ (84). 11. The values observed in step 11 equal independent calculation of ASP II Equations [2.8-12 & 13], ICOUNT = 144, and independent calculation of ASP II Equations [2.8-16] and [2.8-17]. 12. The value calculated in step 12 equals STRFRQ (84). 13. The values observed in step 13 equal ICOUNT = 239 and independent calculation of ASP II Equations [2.8-16] and [2.8-17], respectively. 14. The value calculated in step 14 equals CTRFRQ - OFFFRQ (72).

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	TEST CASE DESCRIPTION
8-4 (Contd)	RESULT: Values of the variable SLOPE in steps 2 and 11 are only half of what hand calculations produce. The value for SLOPE in step 2 is positive in the code, but negative from hand calculations. The code is correct and the negative sign in ASP II Equation [2.8-10] should be deleted. In step 9, the number of counts at the end of the upsweep midpoint, 150, doesn't agree with that calculated by hand, 167. In addition, the upsweep stop frequency in step 10 is at the center frequency, f_c instead of $f_c + \Delta f$ as it should be. In step 11, the number of counts at the beginning of the downsweep, 151, doesn't agree with that calculated by hand, 144. The start frequency of the downsweep calculated in step 12, f_c doesn't equal $f_c + \Delta f$ as it should. In step 13, the number of counts at the end of the downsweep, 311, doesn't agree with that calculated by hand, 239.
8-5	OBJECTIVE: Verify correct calculation of the sum and difference channel complex voltages at the victim radar receiver (tracking radar) for the amplitude modulation ECM technique. PROCEDURE: 1. Start ESAMS, and observe (in the Subroutine BEMEXC) the value of IONFLG returned from the call to the Subroutine GETWOB. 2. Observe the value of VALUE(2) in the Subroutine BEMEXC. 3. Continue execution and verify that the program steps through the subroutines BEMGRM and BEMOUT. 4. Observe in the Subroutine BEMGRM the values of SGSV(1), SGDVA(1), and SGDVE(1). 5. Observe in the Subroutine BEMGRM the values of SGSV(2), SGDVA(2), and SGDVE(2)returned from the call to BEMOUT and compare them to corresponding values observed in step 4. 6. Continue execution until the first time step greater than the pulse off time, TOFF, and observe the values of TIMNOW and TOFF in the Subroutine GETWOB. 7. Observe in the Subroutine BEMEXC the value of IONFLG returned from the call to the Subroutine GETWOB. 8. Observe the value of VALUE(2) in the Subroutine BEMEXC. 9. Continue execution and verify that the program steps through the subroutines BEMGRM and BEMOUT. 10. Observe in the Subroutine BEMGRM the values of SGSV(2), SGDVA(2), and SGDVE(2) returned from the call to BEMOUT and compare them to corresponding values observed in step 5. VERIFY: 1. The value observed in step 1 is equal to 1. 2. The value observed in step 2 is equal to 316.2. 3. Execution transfers from BEMEXC to BEMGRM, BEMOUT and back to BEMGRM in step 3. 4. The variable values observed in step 5 is unique and not equal to that observed in step 4. 5. The value of TIMNOW is slightly larger than TOFF as observed in step 6. 6. The value observed in step 7 is equal to 0. 7. The value observed in step 8 is equal to 0. 8. Execution transfers from BEMEXC to BEMGRM, BEMOUT and back to BEMGRM in step 9. 9. The values of SGSV, SGDVA, and SGDVE observed in step 10 are all equal to zero since the pulse is turned off at this time step. Also verify that SGSV is not equal to what was calculated for this variable in step 5 (it should be non-zero for that step). RESULT: OK

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	TEST CASE DESCRIPTION
8-6	OBJECTIVE: Verify correct calculation of the decoy target geometry for the terrain bounce ECM technique. PROCEDURE: 1. Run ESAMS, and observe in Subroutine ATJCON the value of IRADFL and the execution path following line 226. 2. Continue execution and observe in Subroutine ATJREF the values of CMTDR, CMDPE, and CMDPA. 3. Observe in Subroutine ATJREF the values of CMDPX, CMDPY, and CMDPZ. 4. Observe in Subroutine ATJREF the values of CMDTX, CMDTY, and CMDTZ. 5. Observe in Subroutine ATJREF the values of TERRZ and ITRNSW, and the execution path following line 129. 6. Observe in Subroutine ATJREF the values of CMDMX, CMDMY, CMDMZ, and CMDMR. VERIFY: 1. The value of IRADFL equals 3 and execution transfers to line 228 in step 1. 2. The values of CMTDR, CMDPE, and CMDPA in step 2 match independent calculation of ASP II Equations [2.8-24], [2.8-25], and [2.8-26]. 3. The values of CMDPX, CMDPY, and CMDPZ in step 3 match independent calculation of ASP II Equations [2.8-27], [2.8-28], and [2.8-29]. 4. The values of CMDTX, CMDTY, and CMDTZ in step 4 match independent calculation of ASP II Equations [2.8-30], [2.8-31], and [2.8-32]. 5. The values of TERRZ and ITRNSW equal 0 and 0, and execution transfers to line 153 in step 5. 6. The values of CMDMX, CMDMY, CMDMZ, and CMDMR in step 6 match independent calculation of ASP II Equations [2.8-36], [2.8-37], [2.8-38], and [2.8-39]. RESULT: The values of CMDPX and CMDPY in step 3 do not match independent calculation of ASP II Equations [2.8-27] and [2.8-28]. This was due to errors in the ASP II documentation and not to errors in the code. Taking the squareroot of the quantity (R_{TD}^2 - Z_T^2) in these equations is required to correct this problem.
8-7	OBJECTIVE: Verify correct calculation of the maximum power at the missile seeker assuming perfect ground reflection for the terrain bounce ECM technique. PROCEDURE: 1. Run ESAMS, and observe in Subroutine ATJBOR the values of CMDAE and CMDAA. 2. Continue execution and observe in Subroutine ATJMPI the values of CMTAG and CMTPI. VERIFY: 1. The values of CMDAE, and CMDAA in step 1 equal -0.9094 and -4.2298×10^{-6}, respectively. 2. The values of CMTAG and CMTPI in step 2 equal 0.0045 and 1.2006×10^{-7}, respectively. RESULT: OK

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	TEST CASE DESCRIPTION
8-8	OBJECTIVE: Verify correct calculation of path loss for the terrain bounce ECM technique. PROCEDURE: 1. Run ESAMS, and observe in Subroutine ATJFRC the value of CRX. 2. Continue execution and observe in Subroutine ATJRSC the value of CMRSC. 3. Observe in Subroutine ATJSDC the values of SIGTH2 and SIGPS2. 4. Observe in Subroutine ATJSDC the value of GN. 5. Observe in Subroutine ATJCON the values of PATH and CMARP. VERIFY: 1. The value of CRX in step 1 matches independent calculation of ASP II Equation [2.8-40]. 2. The value of CMRSC in step 2 matches independent calculation of ASP II Equation [2.8-41]. 3. The squareroot of the values for SIGTH2 and SIGPS2 in step 3 match independent calculation of ASP II Equations [2.8-43] and [2.8-44]. 4. The value of GN in step 4 matches independent calculation of ASP II Equation [2.8-42]. 5. The value of PATH in step 5 matches independent calculation of ASP II Equation [2.8-45] and the value of CMARP in this step equals 1.8633×10^{-8}. RESULT: The squareroot of the variable SIGTH2 in step 3 did not match independent calculation of ASP II Equation [2.8-43]. This was due to errors in the ASP II documentation and not to errors in the code. The constant 0.41 should be replaced with 0.42 and the variables φ_R and φ_T should be replaced by θ_R and θ_T, respectively, for ASP II Equation [2.8-43] to be correct. Complete verification of the variable GN in step 4 was not possible due to the omission of an equation (or other definition) for $\sigma_{\alpha 2}$ (SIGAL in the code) from Design Element 2.8-6 of the ASP II.

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	TEST CASE DESCRIPTION
8-9	OBJECTIVE: Verify correct calculation of phase, doppler and range for the terrain bounce ECM technique. PROCEDURE: 1. Run ESAMS, and observe in Subroutine ATJSIG the value of returned for the function PHDEL in the equation for FAC3PT. 2. Continue execution and observe in Subroutine ATJDGP the values of XDMDOT, YDMDOT, and ZDMDOT. 3. Observe in Subroutine ATJDGP the values of XDTDOT, YDTDOT, and ZDTDOT. 4. Observe in Subroutine ATJDGP the values of RDMDOT and RDTDOT. 5. Observe in Subroutine ATJDGP the value of CMDDF. 6. Continue execution and observe in Subroutine ATJCON the value of RTSI(3). VERIFY: 1. The value of PHDEL in step 1 matches independent calculation of ASP II Equation [2.8-46]. 2. The values of XDMDOT, YDMDOT, and ZDMDOT in step 2 match independent calculation of ASP II Equations [2.8-47], [2.8-48], and [2.8-49]. 3. The values of XDTDOT, YDTDOT, and ZDTDOT in step 3 match independent calculation of ASP II Equations [2.8-50], [2.8-51], and [2.8-52]. 4. The values of RDMDOT and RDTDOT in step 4 match independent calculation of ASP II Equations [2.8-53] and [2.8-54]. 5. The value of CMDDF in step 5 matches independent calculation of ASP II Equation [2.8-55]. 6. The value of RTSI(3) in step 6 matches independent calculation of ASP II Equation [2.8-56]. RESULT: The value returned in step 1 was negative while hand calculations returned an identical value, except that it was positive. It is not clear if the equation in the ASP II or the code is the correct one. The developer should review this equation and correct either the documentation or the code. In step 3, ASP II Equations [2.8-50], [2.8-51] and [2.8-52] are the same, i.e., they are all for the decoy-target relative velocity X-component. The latter two equations should be modified to calculate the relative velocities for the Y and Z-components as was probably intended. The value of RDTDOT in step 4 does not match independent calculation of ASP II Equation [2.8-54]. This is due to an error in the ASP II documentation and not to an error in the code. The variable R_{TM} in the ASP II equation needs to be replaced by R_{TD} to correct this discrepancy.

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	TEST CASE DESCRIPTION
8-10	OBJECTIVE: Verify correct calculation of complex voltages at the missile seeker for the terrain bounce ECM technique. PROCEDURE: 1. Run ESAMS, and observe program execution path through the Subroutines BEMGRM, BEMSEN, BEMSVL, BEMEXC, BEMOUT, and ATJCON. 2. Observe in Subroutine ATJSIG the first nonzero values of GSUM, GDIFAZ, and GDIFEL. 3. Observe in Subroutine ATJSIG the values of SUMDCY, DF1DCY, and DF2DCY. 4. Continue execution and observe in Subroutine BEMGRM the values of SGSV(3), SGDVA(3), and SGDVE(3). 5. Observe in Subroutine BEMGRM the values of SGSV(1), SGDVA(1), SGDVE(1), SGSV(2), SGDVA(2), and SGDVE(2). 6. Continue execution until program completion. VERIFY: 1. The execution path in step 1 flows through each of these subroutines in consecutive order. 2. The values of GSUM, GDIFAZ, and GDIFEL in step 2 are not equal to zero. 3. The values of SUMDCY, DF1DCY, and DF2DCY in step 3 match values of SGSV(3), SGDVA(3), and SGDVE(3) observed in step 4. 4. The values of SGSV(3), SGDVA(3), and SGDVE(3) in step 4 are unique and do not match the values of either SGSV(1), SGDVA(1), and SGDVE(1), or SGSV(2), SGDVA(2), and SGDVE(2) observed in step 5. 5. The values of SGSV(1), SGDVA(1), and SGDVE(1), and SGSV(2), SGDVA(2), and SGDVE(2), observed in step 5 are unique and do not match each other. 6. Step 6 is completed without any fatal errors. RESULT: OK
8-11	OBJECTIVE: Verify correct calculation of complex voltages at the missile seeker for the cross-eye jamming technique. PROCEDURE: 1. Run ESAMS, and observe program execution path through the Subroutines BEMGRM, BEMSEN, BEMTVL, BEMEXC, BEMOUT, and BEMSET. 2. Continue execution and observe in Subroutine BEMGRM the values of SGSV(1), SGDVA(1), SGDVE(1), SGSV(2), SGDVA(2), and SGDVE(2) after return from call to BEMOUT. 3. Continue execution until program completion. VERIFY: 1. The execution path in step 1 flows through each of these subroutines in consecutive order. 2. The values of SGSV(1), SGDVA(1), and SGDVE(1), and SGSV(2), SGDVA(2), and SGDVE(2), observed in step 2 are unique and do not match each other. 3. Step 3 is completed without any fatal errors. RESULT: OK

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	**TEST CASE DESCRIPTION**
8-12	OBJECTIVE: Verify correct calculation of complex voltages at the missile seeker for the RGPO ECM technique. PROCEDURE: 1. Run ESAMS for case of RGPO jamming and break at BEMGRM when JAMIT =1. Observe program execution path through the Subroutines BEMGRM, BEMSEN, BEMTVL, BEMEXC, BEMOUT, and BEMSET. 2. Continue execution and observe in Subroutine BEMGRM the values of SGSV(1), SGDVA(1), SGDVE(1), SGSV(2), SGDVA(2), and SGDVE(2) after return from call to BEMOUT. 3. Continue execution until program completion. VERIFY: 1. The execution path in step 1 flows through each of these subroutines in consecutive order. 2. The values of SGSV(1), SGDVA(1), and SGDVE(1), and SGSV(2), SGDVA(2), and SGDVE(2), observed in step 2 are unique and do not match each other. 3. Step 3 is completed without any fatal errors. RESULT: OK

TABLE 2.8-6. Software Test Cases for Onboard Deceptive ECM FE. (Contd.)

Test Case ID	TEST CASE DESCRIPTION
8-13	OBJECTIVE: Verify correct calculation of complex voltages at the missile seeker for the VGPO ECM technique. PROCEDURE: 1. Run ESAMS for case of VGPO jamming and break at BEMGRM when JAMIT =1. Observe program execution path through the Subroutines BEMGRM, BEMSEN, BEMTVL, BEMEXC, BEMOUT, and BEMSET. 2. Continue execution and observe in Subroutine BEMGRM the values of SGSV(1), SGDVA(1), SGDVE(1), SGSV(2), SGDVA(2), and SGDVE(2) after return from call to BEMOUT. 3. Continue execution until program completion. VERIFY: 1. The execution path in step 1 flows through each of these subroutines in consecutive order. 2. The values of SGSV(1), SGDVA(1), and SGDVE(1), and SGSV(2), SGDVA(2), and SGDVE(2), observed in step 2 are unique and do not match each other. 3. Step 3 is completed without any fatal errors. RESULT: OK
8-14	OBJECTIVE: Verify correct calculation of complex voltages at the missile seeker for the RVGPO ECM technique. PROCEDURE: 1. Run ESAMS for case of RVGPO jamming and break at BEMGRM when JAMIT =1. Observe program execution path through the Subroutines BEMGRM, BEMSEN, BEMTVL, BEMEXC, BEMOUT, and BEMSET. 2. Continue execution and observe in Subroutine BEMGRM the values of SGSV(1), SGDVA(1), SGDVE(1), SGSV(2), SGDVA(2), and SGDVE(2) after return from call to BEMOUT. 3. Continue execution until program completion. VERIFY: 1. The execution path in step 1 flows through each of these subroutines in consecutive order. 2. The values of SGSV(1), SGDVA(1), and SGDVE(1), and SGSV(2), SGDVA(2), and SGDVE(2), observed in step 2 are unique and do not match each other. 3. Step 3 is completed without any fatal errors. RESULT: OK

2.8.5 Conclusions and Recommendations

2.8.5.1 Code Discrepancies

In general, the coded algorithms implement the design criteria correctly although several serious discrepancies were uncovered during verification of the Onboard Deceptive ECM FE for ESAMS 2.7. For the first discrepancy, the slopes of the upsweep and downsweep for the sawtooth-type sweep case are calculated incorrectly. The frequency sweep repeat period in the denominators of these equations should be divided by two to account for the fact that with a sawtooth sweep type, the upsweep and downsweep periods are only half of the total sweep period. This omission results in slopes that are only half of what they should be. Multiplying these equations by two will correct this discrepancy.

For the second discrepancy, the number of pulses required to complete a sweep in the code doesn't agree with the number calculated using the ASP II equations. In addition, when the upsweep is completed for a sawtooth-type waveform, its stop frequency is only at the center frequency instead of being at the center frequency plus the offset frequency. This is due to the first discrepancy, however, if this discrepancy is corrected, the start and stop frequencies of the upsweep are correct, but the start frequency of the downsweep is down shifted by twice the offset frequency. In other words, the start frequency of the downsweep starts at the center frequency minus the offset frequency instead of starting at the center frequency plus the offset frequency. This problem and the one regarding the incorrect pulse count per sweep are related to the unusual way the ASP II equations are implemented and can be solved by reformulating equations for the code variables DISCR and ZROTIM.

For the third discrepancy, a one is inadvertently added to the target-decoy slant range in the denominator of the argument for the decoy elevation angle calculations in both the subroutine ATJREF and in ASP II Equations [2.8-25] and [2.8-35]. This one appears to be added to prevent a singularity condition that would occur if the target-decoy slant range were to be less than the target height in the numerator of the argument. This would result in the code attempting to take the arcsine of an argument larger than one which would cause a fatal execution error. Rather than introducing this small error into the equation to prevent such singularities, it would preferable to test this argument and use one for the argument should the real argument actually be greater than one.

The fourth discrepancy is essentially the same problem described in the third discrepancy, but applied to different equations. This discrepancy regards the calculation of the decoy elevation and azimuth angles, CMDAE and CMDAA, in the Subroutine ATJBOR. These equations involve taking the arcsine and arctangent of their respective arguments. Like before, it appears that a one was added to the denominator of their arguments to prevent singularity conditions.

The fifth discrepancy is also similar to the third and fourth discrepancies, but doesn't involve any trigometric functions. This discrepancy regards the calculation of the target-missile altitude, HTHM, in the Subroutine ATJSDC. It appears that a one was added to the missile position Z-component in the equations denominator to prevent the target-missile ratio from exceeding one. If this ratio were to exceed one it could possibly result in the spatial distribution coefficient, GN, exceeding one which is theoretically impossible. Rather than introducing this small error into the equation to prevent such incongruities, it would preferable to test the equation for HTHM and use a ratio of one should the real ratio actually be greater than one.

2.8.5.2 Code Quality and Internal Documentation

The quality of the code for the Onboard Deceptive ECM FE in ESAMS 2.7 is generally good. Nonetheless, variable declarations are missing from the subroutine BEMANT and should be added. In addition, the subroutines BEMGRM, BEMTVL, BEMSVL, BEMANT, ECMINI, GETWOB, ATJCON, ATJI, ATJREF, and ATJSIG contain unnecessary INCLUDE statements for common blocks as well. Several code quality discrepancies were discovered in the subroutine BEMOUT. The first involves redundant calls to the subroutine GYRATE in both branches of an IF statement. By moving this call to precede the IF statement, this can be accomplished with only one line of code instead of two.

The second code quality discrepancy in BEMOUT regards converting the results from ASIN and ATAN2 functions from radians to degrees with the variable R2D. Use of the functions ASIND and ATAND2 instead of ASIN and ATAN2 would return values in degrees rather than radians. This should be more efficient.

Internal documentation was fair with numerous problems addressed in Table 2.28-5. The comments preceding the array element RADVLU(2) in the subroutines BEMTVL and BEMSVL are wrong and need to be corrected. There were numerous variable description errors/omissions in many subroutines that need addressing. Most of the headers for these subroutines don't have completely documented information regarding the subroutine's author, version #, abstract, and purpose/technical description as well.

2.8.5.3 External Documentation

The external documentation is good for the subjects discussed in the User's, Advanced User's, Analyst's, and ECM Manuals. A programmer's manual should be developed to describe the software implementation of the theory used to develop ESAMS. Other than choosing which jammer/technique to use, there is no direct user interface to the Onboard Deceptive ECM FE, therefore, it is not discussed in the User's nor the Advanced User's Manuals. The ECM Manual contains an adequate, although upper level explanation of Onboard Deceptive ECM methodology.

The external documentation is also generally good for Conceptual Model Specification regarding the Onboard Deceptive ECM FE despite several errors that were discovered. Most of these are minor mathematical errors and are described in the following paragraphs. The equation for the variable SLOPE on line 75 of the subroutine GETWOB does not match ASP II Equation [2.8-10]. The equation in the code is positive, while that in the documentation is preceded by a negative sign. This calculation is for the slope of the upsweep for the sawtooth frequency sweep type. The equation in the documentation is identical with ASP II Equation [2.8-12], which is the slope of the downsweep for the sawtooth frequency sweep type. Since the slope of the frequency upsweep must be positive, the code is correct and the documentation is erroneous. Remove the minus sign from ASP II Equation [2.8-10] to correct this discrepancy.

The design elements for Complex Voltage from Amplitude Modulation Jamming and Terrain Bounce Geometry are both labeled as 8-3. Change the designation for the Terrain Bounce Geometry to be Design Element 8-4 and increment all subsequent design elements by one. All design element references in this document have accounted for this error.

ASP II Equations [2.8-25] and [2.8-35] show a one being added to the denominator of the arcsine's argument. This doesn't make sense analytically and appears to have been done to prevent singularity conditions that would arise if the argument were to exceed one. Details of his problem can studied by reviewing code discrepancy D3 in Table 2.8-4.

The equations for the variables CMDPX and CMDPY on lines 120 and 121 of the Subroutine ATJREF do not match ASP II Equations [2.8-27] and [2.8-28]. The equations in the code are correct and taking the squareroot of the quantity $(R_{TD}^2 - Z_T^2)$ in the ASP II equations for X_D and $Y_D$will correct this discrepancy.

The equation for the variable CMTDR on line 147 of the Subroutine ATJREF doesn't match ASP II Equation [2.8-34]. The equation in the code is correct and the variable R_{TDZ} in the ASP II equation needs to be squared to correct this discrepancy.

No equation (or other definition) exists in the documentation to define the variable $\sigma_{\alpha 2}$, which is used in ASP II Equation [2.8-42]. This equation needs to be defined in the ASP II in order to verify its correct implementation in the equation for SIGAL on line 85 of the Subroutine ATJSDC.

The equation for the variable SIGTH2 on line 73 of the Subroutine ATJSDC doesn't match ASP II Equation [2.8-43]. The equation in the code is correct (although it is the squared result of the ASP equation) and the variables φ_R and φ_T need to be replaced by θ_R and θ_T, respectively, in the ASP II equation to correct this discrepancy. In addition, the constant 0.41 should be replaced by 0.42 to complete the corrections to ASP II Equation [2.8-43].

The equation for the variable PHDEL on line 94 of the Subroutine ATJSIG doesn't match ASP II Equation [2.8-46]. It is not clear if the error is in the documentation or the code. The equation in the code results in a negative phase delay, while that in the documentation shows this quantity to be positive, but of equal value.

The calculations for the decoy to target relative velocity components shown in the ASP II Equations [2.8-50], [2.8-51], and [2.8-52] are redundant and all for the X-component only. The latter two equations should be modified to be for the Y and Z-components as was probably initially intended.

The equation for the variable RDMDOT on line 103 of the Subroutine ATJDGP does not match ASP II Equation [2.8-54]. The equation in the code is correct and replacing the variable R_{TM} with R_{TD} in the ASP II equation will correct this discrepancy.

It was difficult to verify the correct implementation of the spatial distribution coefficient equation of Design Element 8-6 (ASP II Equation [2.8-42]) due to the lack of the source reference document. This coefficient is discussed in the Unique ECM Techniques subsection of Section 2 in the ECM Manual, but the final equation implemented in the code is not shown. A statement was made saying that Hughes Aircraft modified the equation derived in the ECM Manual to match empirical data, but the resulting equation was never shown. The ECM Manual should be modified to show the equation that was actually implemented in subroutine ATJSDC. Efforts should be made to update the CMS for Onboard Deceptive ECM to correct the documentation errors described in this report before including it in the ASP II document.

The VSDR for Onboard Deceptive ECM shows the terrain bounce subroutines, ATJBOR, ATJCON, ATJDGP, ATJFRC, ATJRSC, ATJSDC, ATJI, ATJMPI, ATJREF, and ATJSIG, being classified as 'UNCLASSIFIED', while the Security Classification Guide for ESAMS 2.7 (Oct. 31, 1995) shows them as being classified as 'SECRET'. This discrepancy should be corrected as necessary by modifying the erroneous document (most likely the VSDR).

It was noted that the subroutine for jamming fuze radars, BEMFVL, was in the VSDR for this FE, but was omitted from the CMS/ASP II document. In addition, it was included in the FE for Onboard Noise ECM because of the possibility of continuous noise jamming. It is recommended that fuze jamming be included (and thus the subroutine BEMFVL) in this FE as well. This is because of the possibility of successfully jamming a fuze radar using a RGPO technique, thus causing premature warhead detonation. No information regarding the feasibility of noise jamming against a fuze radar could be found, thus it is recommended that this capability be deleted from the Onboard Noise ECM FE. A fuze radar jamming capability would also be appropriate to include in the FE for Off-Board Deceptive ECM.

DRAFT